



TEKNILLINEN TIEDEKUNTA

KONEOPPIMISEN MENETELMIÄ

Joni Saarikettu

PROSESSITEKNIIKAN KOULUTUSOHJELMA

Kandidaatintyö

Lokakuu 2019



TEKNILLINEN TIEDEKUNTA

KONEOPPIMISEN MENETELMIÄ

Joni Saarikettu

Ohjaajat: Aki Sorsa, Mika Ruusunen

PROSESSITEKNIIKAN KOULUTUSOHJELMA

Kandidaatintyö

Lokakuu 2019

TIIVISTELMÄ OPINNÄYTETYÖSTÄ

Oulun yliopisto Teknillinen tiedekunta

Koulutusohjelma (kandidaatintyö, diplomityö) Prosessitekniikan koulutusohjelma		Pääaineopintojen ala (lisensiaatintyö)	
Tekijä Saarikettu, Joni		Työn ohjaaja yliopistolla Aki Sorsa, TkT & Mika Ruusunen, Prof.	
Työn nimi Koneoppimisen menetelmiä			
Opintosuunta Säätö- ja systeemitekniikka	Työn laji Kandidaatintyö	Aika Lokakuu 2019	Sivumäärä 30
<p>Tiivistelmä</p> <p>Työssä oli tavoitteena kirjallisuusselvityksenä käydä läpi koneoppimista yleisellä tasolla menemättä koneoppimisen matemaattiseen puoleen tai koodaamiseen, mikä kylläkin on tärkeä osa koneoppimista. Työssä kuitenkin käytiin vain pintaraapaisu koneoppimisesta, mutta kuitenkin sen verran kattavasti, että tämän työn luettuaan ymmärtää mistä koneoppimisesta ja siihen liittyvissä menetelmissä on kyse. Työssä käytettiin paljon esimerkkejä, joilla pyrittiinkin luomaan selkeämpää kuvaa menetelmistä.</p> <p>Johdannossa avataan koneoppimista määritelmänä ja käydään läpi koneoppimiseen liittyviä olennaisia termejä. Koneoppimisen määritelmää selventämään käydään läpi tammipeliä, jotta hankalalta vaikuttavat termit selkeytyvät lukijalle. Johdannon jälkeen paneuduttiin enemmän koneoppimisen menetelmiin ja annettiin käytännön esimerkkejä jokaisen menetelmän kohdalla. Menetelmät, joihin työssä paneudutaan, ovat käytetyimpiä ja tunnetuimpia ja kirjoittajan mielestä olennaisia koneoppimiseen liittyen. Jotta työ ei jäisi pelkästään pintapuoliseksi raapaisuksi, käydään läpi koneoppimisalgoritmien optimointia ja siihen liittyviä asioita kuten kohdefunktiota tarkemmin. Lisäksi pureudutaan optimoinnin haasteisiin, jotka liittyvät olennaisesti koneoppimisen tehokkuuteen.</p> <p>Työssä käydään myös tarkemmin läpi muutamaa sovellusta, jotka on saavutettu koneoppimisen avulla. Pari näistä sovelluksista on suurelle yleisölle hyvinkin tunnettuja ja monelle lähes jokapäiväisessä käytössä. Sitten on myös pari vähän tuntemattomampaa sovellusta käyty läpi vastapainoksi. Siltikin voi tulla yllätyksenä, että juuri koneoppimisella on luotu nämä sovellukset kuten sähköpostin roskapostisuodatin. Työn loppupuolella on kirjoittajan omaa pohdintaa liittyen tekoälyn mahdollisuuksiin lähitulevaisuudessa sekä mahdollisia uhkakuvia nykyisen yhteiskunnan kannalta.</p>			
Muita tietoja			

SISÄLLYSLUETTELO

TIIVISTELMÄ OPINNÄYTETYÖSTÄ.....	2
SISÄLLYSLUETTELO.....	3
1 Johdanto	4
2 Oppimismenetelmät	6
2.1 Yleisesti oppimismenetelmistä.....	6
2.1.1 Ohjattu oppiminen	6
2.1.2 Ohjaamaton oppiminen.....	7
2.1.3 Reinforcement Learning	7
2.2 Pääospuuoppiminen (Decision tree learning)	8
2.3 Naiivi Bayesin luokitin	10
2.4 Neuroverkot.....	10
2.5 Tukivektorikone	12
2.6 Itseohjautuvakartta	13
2.7 Klusterointi.....	14
3 Oppimisen optimointi	16
3.1 Kohdefunktio.....	16
3.2 Optimointimenetelmät.....	16
3.2.1 Particle Swarm Optimization.....	17
3.2.2 Geneettinen algoritmi	18
3.2.3 Differential Evolution.....	18
3.3 Optimoinnin ongelmat	19
3.3.1 Ylioppiminen	19
3.3.2 Alioppiminen	20
4 Sovellukset	21
4.1 Hämmontunnistus.....	21
4.2 Lääketieteellinen diagnoosi.....	22
4.3 Sähköpostin roskapostisuodatin	22
4.4 Mineraalien erottaminen hiekan seasta koneäön avulla.....	23
5 Tulevaisuuden visioita tekoälystä	24
6 Yhteenveto	26
Lähdeluettelo.....	27

1 JOHDANTO

Yksinkertaisimmillaan koneoppimisella tarkoitetaan, kun tietokoneen tai tietokoneohjelman suorituskyky halutussa tehtävässä paranee oppimisen tuloksena. [1] Tietokone oppii siis itsenäisesti tekemällä toistoja lukemattomia kertoja ilman varsinaista ohjelmointia, miten toimia kussakin tilanteessa. [2] Itsenäistä oppimista varten tarvitaan oppimismenetelmä, kohdefunktio ja harjoittelukokemus. Oppimismenetelmä tarkoittaa käytännössä optimointialgoritmia, joka pyrkii muokkaamaan systeemin toimintaa kohti kohdefunktion ja harjoittelukokemusten määrittelemää optimaalista ratkaisua. Oppimista arvioidaan jonkin suorituskykymittarin avulla, kuinka se kehittyy harjoittelun tuloksena. Kun tehtävä, suorituskykymittari ja harjoittelutapa tai kokemus on valittu, siirrytään kohdefunktion valintaan. [1]

Kohdefunktiolla tarkoitetaan yksinkertaistetusti oppimistehtävän tavoitetta. Kohdefunktion valinta on tärkeä, koska se määrittää miten ja mitä tietokone oppii tai toisin sanoen kuinka hyvin se on optimoitu. Oikean kohdefunktion löytäminen oppimisen kannalta on yksi avaintekijöistä. Jos kohdefunktio valitaan väärin, oppiminen hidastuu, tai on vääränlaista eikä tietokone pysty suoriutumaan tehtävästä tarpeeksi hyvin tai ollenkaan. [1]

Koneoppimisessa tärkeä termi on yleistävyys (englanniksi generalizability). Sillä tarkoitetaan sitä, kuinka hyvin algoritmi pystyy ennustamaan sellaisten näytteiden tai datan käyttäytymistä, jota ei ole käytetty opetusdatassa. Eli toisin sanoen kuinka hyvin algoritmi pystyy toimimaan tilanteissa, joita sille ei ole aikaisemmin tullut vastaan. [3]

Tom Mitchell esittää kirjassaan Machine Learning esimerkkinä koneoppimisesta, kun tietokonetta opetetaan pelaamaan itsenäisesti tammea. Tehtäväksi valitaan pelata tammea, suorituskyvyn mittariksi valitaan kuinka monta prosenttia peleistä tietokone voittaa vastustajiaan vastaan ja harjoittelukokemukseksi valitaan tietokone pelaamaan itseään vastaan pelejä. Tarpeeksi monella toistolla tietokoneen pitäisi oppia tunnistamaan tietyt tilanteet ja oppia mikä siirto tilanteessa olisi kaikista järkevintä, kun tavoitteena on pelin voittaminen. Kohdefunktio tässä esimerkkitapauksessa on valita paras siirto sallituista siirroista tavoitteena pelin voittaminen. [1]

Tom Mitchell esittää kirjassaan myös, kuinka tietokone voidaan opettaa tunnistamaan käsinkirjoitettua sanoja. Tässä tapauksessa tehtäväksi valitaan käsinkirjoitettujen sanojen tunnistaminen kuvista ja sanojen luokittelu. Suorituskyvyn mittarina tekoälyn oppimisesta kertoo, kuinka monta prosenttia sanoista on tunnistettu ja luokiteltu oikein. Harjoittelukokemuksena käytetään tietokantaa käsinkirjoitetuista sanoista, jotka on jollakin perusteella luokiteltu valmiiksi. [1]

Koneoppimista käytetään tapauksissa, joihin tietokone soveltuu ihmistä paremmin kuten yllä mainittu kuvantunnistus. Tietokone kykenee oikein opetettuna tunnistamaan haluttua dataa suuresta otannasta huomattavasti ihmistä nopeammin. [1] Koneoppiminen on todella tehokas tapa silloin kuin jokin tehtävä on hankala ratkaista ”manuaalisella” ohjelmoinnilla. Eli kun ohjelmoitava kohde on liian monimutkainen ohjelmoida antamalla ohjelmalle selvät ohjeet tai jos ohjelmoija ei osaa yksinkertaisesti ohjelmoida ohjelmaa toimimaan kuten haluaa. [2]

2 OPPIMISMENETELMÄT

2.1 Yleisesti oppimismenetelmistä

Oppimismenetelmät voidaan karkeasti jakaa kahteen eri alakategoriaan, jotka ovat ohjattu oppiminen ja ohjaamaton oppiminen. Jos aihetta alkaisi käydä syvällisemmin läpi, kaikki koneoppimisen menetelmät eivät mahdu näiden kahden menetelmän alle kuitenkaan vaan näidenkin oppimismenetelmien alla on alakategorioita. Keskitytään näihin kahteen pääkategoriaan tässä työssä, jotta aiheet ovat helpommin sisäistettävissä. [2] Lisäksi käydään läpi yleisimpiä ja suosituimpia käytössä olevia koneoppimisalgoritmeja ja avataan näiden pääperiaatteet ja toimintamallit menemättä sen kuitenkaan matemaattisiin malleihin tai koodaamiseen.

2.1.1 Ohjattu oppiminen

Ohjatussa oppimisessa on kyse siitä, kun opetusdatasta tiedetään haluttu lopputulos ja yritetään saada ohjelma opetusdatan avulla pääsemään haluttuun lopputulokseen. Nimitys ohjattu oppiminen tulee siitä, ettei tietokone pysty itsenäisesti päättämään oikeita vastauksia vaan ohjaajan täytyy kertoa, mikä on oikea vastaus. Lopulta kone pystyy itsenäisesti päättämään oikeisiin lopputuloksiin, mutta tämä vaatii lukemattomia toistoja. [4]

Yleisesti ohjatun oppimisen ongelmat voidaan jakaa kahteen alakategoriaan: luokitteluun ja regressioon. Luokitteluongelmissa on kyse muuttujien luokittelemisesta ennalta määritettyihin kategorioihin. Esimerkiksi roskapostisuodattimet toimivat tällä periaatteella. Algoritmi tulkitsee opetusdatan perusteella, onko saapuva sähköpostia roskapostia vai ei ja sen perusteella ohjaa sähköpostiviestin roskapostikansioon. [3]

Esimerkkinä ohjatusta oppimista käy verkkosivu (<http://kannattaakokauppa.fi/#/fi/>), jolla pystyy tarkastelemaan kannattaako asuntokauppa tietyllä alueella Suomessa pohjautuen alueella tapahtuneisiin asuntokauppoihin viimeisten vuosien aikana. Palvelu kertoo trendin asuinalueittain ja kuinka paljon on asuinalueen keskimääräinen asuinneliöhinta toteutuneissa kaupoissa. Opetusdatana on käytetty, sitä kuinka monta asuntokauppaa on

tapahtunut viimeisten vuosien aikana ja toteutuneiden asuntokauppojen asuinneliön keskihintaa. [5]

Tästä päästäänkin yhteen koneoppimisen ongelmaan; siihen ettei tämäkään ohjelma pysty luotettavasti kertomaan jatkuuko tuleva trendi tietyllä asuinalueella. Neliöhinnan arvioiminen asuinalueella on kuitenkin helpottunut tämän palvelun avulla. [4]

2.1.2 Ohjaamaton oppiminen

Ohjaamaton oppiminen eroaa ohjatusta oppimisesta siinä, ettei opetusdatassa ole valmiita tuloksia olemassa. On vain datajoukko jostain asiasta. Ohjaamattomalla oppimisella koneoppimisalgoritmi kykenee opetusdatasta tekemään johtopäätöksiä ja näkemään syötetyssä datassa yhtäläisyyksiä ja samalla myös eroavaisuuksia. Ohjaamattomassa oppimisessä on kyse monesti siitä, että kone löytää jonkinlaisen rakenteen datasta. [4]

Ohjaamaton oppiminen voidaan myös nähdä sillä tavalla, ettei oppimisprosessissa ole mukana ohjaajaa, joka antaisi palautetta oppimisprosessin aikana vaan oppiminen on itsenäistä. Tästä tuleekin nimitys tälle oppimistavalle. [6]

Matemaattisesti ajatellen ohjaamattomassa oppimisessä on kyse siitä, kun sinulla on vain lähtötiedot eikä vastaavia ulostulomuuttujia ole tiedossa. [4]

2.1.3 Reinforcement Learning

Reinforcement learning-termillä tarkoitetaan oppimista, joka on periaatteessa välimuoto ohjatun oppimisen ja ohjaamattoman oppimisen välillä. Reinforcement learning perustuu jatkuvaan vuorovaikutukseen ympäristön ja oppimisjärjestelmän (agentti) välillä. Oppimisjärjestelmä suorittaa toiminnon ja näin koneoppimisalgoritmi oppii ympäristön vasteesta mitä tapahtuu tietyssä tilanteessa ja kuinka toimia vastaavanlaisessa tilanteessa tulevaisuudessa. Ohjattuun oppimiseen verrattuna ohjaaja on korvattu tässä tapauksessa kriitikolla, joka on integroitu oppimisjärjestelmään. Se antaa palautetta, joka on koneoppimisalgoritmin oppimisen kannalta hyvin tärkeää. Tämän palautteen perusteella oppimisjärjestelmä oppii toimimaan kuten sen halutaan toimivan vastaavanlaisissa tilanteissa. [6]

Reinforcement learning käytetään tilanteissa, joissa tarvitaan moniulotteista oppimista, eivätkä tilanteet ole välttämättä yksiselitteisiä. Mahdollisesti myös oppimisjärjestelmän saama palaute, ratkaisun oikeellisuudesta, tulee viiveellä, joka taas hankaloittaa oppimistilannetta. Hyvänä esimerkkinä käy itseohjautuvien autojen opettaminen, joiden opettaminen ei onnistu mustavalkoisesti ”joko/tai” periaatteen mukaan vaan pitää katsoa tarkemmin tilannetta kokonaisuutena. [4]

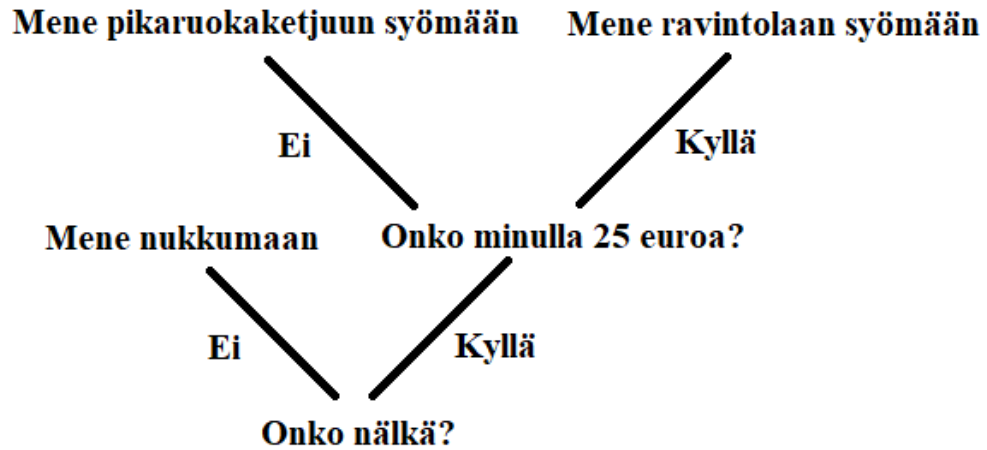
2.2 Päättöspuuoppiminen (Decision tree learning)

Päättöspuuoppiminen (englanniksi Decision tree learning) on yleisesti käytössä oleva koneoppimisen oppimismenetelmä. Päättöspuuoppiminen kuuluu ohjatun oppimisen alle ja on pohjimmiltaan yksinkertainen menetelmä. [7]

Päättöspuilla yleisesti tarkoitetaan päätöksentekomenetelmää, joka perustuu useimmiten kahden vaihtoehdon kysymyksiin, jotka johtavat eteenpäin päätöksenteossa tai vaihtoehtoisesti lopputulokseen. Päättöspuuta voidaan käyttää sekä regressio-, että luokitteluongelmien ratkaisemiseen. Kun päättöspuuta käytetään regressio-ongelman ratkaisuun, sitä kutsutaan regressiopuiksi. Vastaavasti kun päättöspuuta käytetään luokitteluongelman ratkaisuun, sitä kutsutaan luokittelupuiksi. [7]

Päättöspuulla tarkoitetaan päätöksentekoa valitsemalla kahdesta vaihtoehdosta toinen, joka päättyy joko seuraavaan oksaan, jossa tehdään seuraava päätös tai vaihtoehtoisesti päätös johtaa ”lehteen” eli toisin sanoen lopputulokseen. Tätä jatketaan niin kauan, kunnes päädytään ”lehteen”. [7]

Alla olevassa kuvassa esiintyvä kysymys ”Onko nälkä?” on päättöspuun juuri, josta lähdetään liikkeelle. Jos vastaus on ”kyllä”, siirrytään seuraavaan kysymykseen oksaa pitkin tai vaihtoehtoisesti jos vastaus on ”ei”, päädytään lehteen, jonka lopputulos on mennä nukkumaan. Vastauksen ollessa ”kyllä” kysytään, ”onko minulla 25 euroa rahaa?” ja riippuen vastauksesta mennään joko ravintolaan syömään tai sitten mennään pikaruokaketjuun syömään, jotka ovat molemmat lopputuloksia tässä tapauksessa. [7]



Kuva 1. Päättöspuumenetelmä

Kuvassa 1 käyty esimerkki on hyvin yksinkertainen verrattuna koneoppimisessa käytettäviin päätöspuihin, mutta kuvaa silti hyvin päätöspuun yksinkertaisuutta päätöksenteossa. Todellisuudessa koneoppimisessa käytetyt päätöspuut ovat huomattavasti monimutkaisempia. Toisaalta jos päätöspuusta tulee liian monimutkainen, sen yleistyskyky heikkenee eikä sillä pystytä niin suurella todennäköisyydellä saavuttamaan haluttua lopputulosta. [7]

Päättöspuuoppimista voidaan käyttää esimerkiksi algoritmille roskapostin suodattamista varten. Opetusdatana voidaan käyttää sähköpostin käyttäjien roskapostiksi merkkauksia sähköpostiviestejä ja suorituskäytön mittarina, kuinka monta prosenttia sähköpostiviesteistä algoritmi osaa lajitella oikeaan kansioon. Päättöspuun oksina voisi toimia vaikkapa uusien vastaanottajien määrä, jonka kasvaessa liian suureksi algoritmi lajittelee sähköpostiviestin roskapostikansioon tai vaihtoehtoisesti algoritmi voi tarkastella sähköpostiviestin pituutta ja sitä kautta tehdä päätöksen perustuen opetusdatasta opittuihin kriteereihin. [7]

2.3 Naiivi Bayesin luokitin

Naiivi Bayesin luokitin on koneoppimisalgoritmi, jonka avulla voidaan luokitella data-aineistoa kahteen tai useampaan luokkaan. Algoritmia opetetaan analysoimaan aineistoja opetusdatan avulla, johon on merkitty valmiiksi oikeat vastaukset. Naiivi Bayesin luokitin luetaan ohjatun oppimisen alle. [8]

Naiivi Bayesin luokitin perustuu oletukseen, että kaikki havainnot data-aineistosta ovat ehdollisesti toisistaan riippumattomia tietyssä luokassa. Tämä on ns. naiivi Bayesin oletus. Todellisuudessa tämä oletus on väärin todella monissa tilanteissa, mutta luokitteluongelmissa sen on huomattu toimivan erittäin hyvin, jonka takia naiivi Bayesin luokitin on yksinkertainen ja yleisesti käytössä oleva luokittelualgoritmi. [9]

Otetaan esimerkiksi roskapostisuodatin selventämään tämän koneoppimisalgoritmin toimintaa. Naiivi Bayesin oletus tässä tapauksessa tarkoittaa sitä, ettei sähköpostiviestissä esiintyvät sanat riipu toisistaan vaan ainoastaan siitä onko sähköpostiviesti roskaposti vaiko ei. Viestissä esiintyvät sanat ovat havaintoja, joiden todennäköisyyksien perusteella roskaposti lajitellaan algoritmin mielestä oikeaan kansioon. Todennäköisyydet sanoille, jotka esiintyvät roskaposteissa perustuvat algoritmille opetettuun dataan. Naiivi Bayesin luokitin lajittelee toisin sanoen viestit niissä esiintyvien sanojen ja niiden todennäköisyyksien perusteella. Onkin hyvin paljon kiinni opetusdatasta, kuinka hyvin optimoitu naiivi Bayesin luokitin on. Opetusdata, jossa on sekä asiallisia viestejä, että roskapostia sopivassa suhteessa toimii parhaiten algoritmin opetukseen ja optimointiin. [8]

2.4 Neuroverkot

Simon Haykin kirjassaan *Neural Networks and Learning Machines* (2009) määrittelee neuroverkon seuraavasti: ”Neuroverkko on massiivisesti rinnakkain hajautettu prosessori, joka koostuu yksinkertaisista prosessiyksiköistä, joilla on luontainen taipumus tallentaa kokemuksen kautta opittua tietoa ja tehdä siitä käytettävää dataa. Se muistuttaa aivoja kahdessa suhteessa:

1. Neuroverkko hankkii tietoa ympäristöstään oppimisprosessin kautta.

2. Neuroneita ja niiden välisiä yhteyksiä käytetään hankitun tiedon tallentamiseen.”
[6]

Yllä olevassa määritelmässä yksinkertaisella prosessiyksiköllä tarkoitetaan neuronia, joka on siis laskennallinen yksinkertainen prosessiyksikkö verrattavissa aivojen vastaaviin neuroneihin. [6]

Toisin sanoen neuroverkoilla tarkoitetaan monimutkaisia, ihmisaivoja mukailevia oppimisalgoritmeja. Neuroverkot koostuvat yksittäisistä neuroneista, jotka muodostavat lukemattomista neuroneista koostuvan kokonaisuuden, joka pystyy laskennallisesti raskaisiin ja vaikeisiin tehtäviin, joihin ei yksinkertaisemmilla algoritmeilla pystytä. [10] Kuitenkin yleisellä tasolla ja erittäin yksinkertaistetusti neuroverkoilla monesti tarkoitetaan konetta, joka matkii aivojen toimintaa tietyssä tehtävässä. [6]

Neuroverkkoja on mahdollista opettaa sekä ohjatusti että ohjaamattomasti. Tämä lisää neuroverkkojen käyttömahdollisuuksia huomattavasti, sillä opetusdatan syöttäminen neuroverkolle on mahdollista monella eri tavalla. [6]

Kun neuroverkkoja opetetaan toimimaan jotain tehtävää varten, niin poiketen monista koneoppimisalgoritmeista, sille ei tarvitse antaa tarkkoja sääntöjä tai ohjeistusta syötettävän datan suhteen. Sen sijaan neuroverkon opetuksessa käytetään laajaa opetusdataa, joissa on valmiit vastaukset. Lukemattomien toistojen jälkeen neuroverkko oppii syötettävästä datasta tunnistamaan mikä on oikea ratkaisu kussakin tilanteessa. Kun neuroverkolle on syötetty tarpeeksi opetusdataa, se pystyy itsenäisesti suoriutumaan täysin uusista tilanteista ja kehittymään entisestään. [10]

Otetaan esimerkiksi kuvantunnistus, jossa on tarkoitus tunnistaa kissa erilaisista kuvista. Normaaleilla koneoppimisalgoritmeilla tämä on lähes mahdoton tehtävä, sillä pitäisi luoda sääntö jokaista erilaista kuvaa varten, mutta neuroverkko kykenee kokemuksen kautta oppimaan tunnistamaan kissan riittävän opetusdatan avulla. Tämä perustuu kuvista löytyviin samankaltaisuuksiin, jotka ovat tulleet sitä vastaan aiemmin. Jokaisen kuvan kohdalla neuroverkko vertaa uutta kuvaa vanhoihin kokemuksiin, joiden avulla se on opetettu. Tämän aikaisempiin kokemuksiin perustuvan mallin perusteella algoritmi

päättelee onko kuvassa kissa vaiko ei. Jokainen uusi tilanne kehittää neuroverkkoa entistä tehokkaammaksi ja lisää todennäköisyyttä tunnistaa kissan kuvasta. [10]

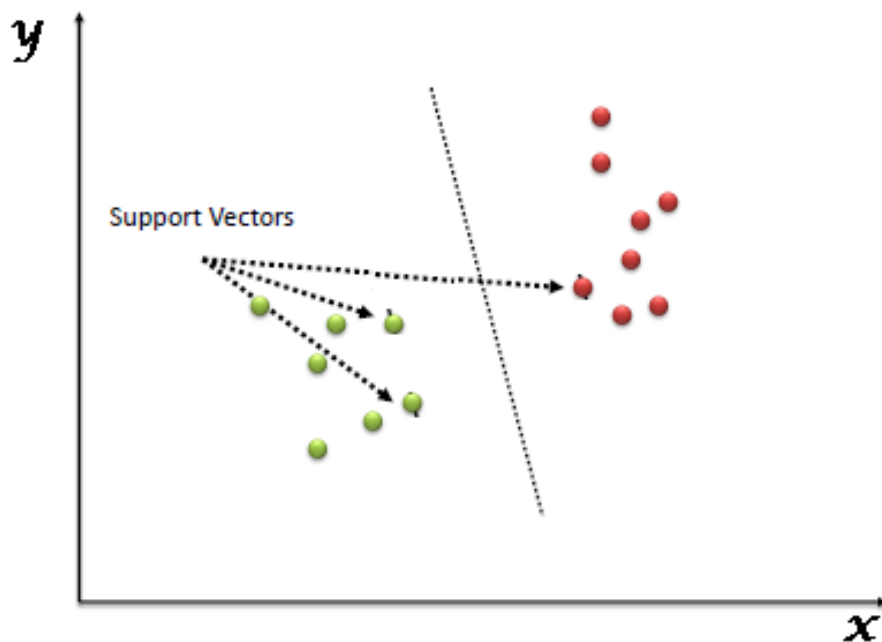
Neuroverkot on valjastettu nykyaikana myös esimerkiksi puheen- ja kasvojentunnistukseen, roskapostinsuodatuksen ja lääketieteellisen diagnoosin tekemiseen. [10]

2.5 Tukivektorikone

Tukivektorikoneet (englanniksi Support Vector Machine, SVM) on lineaarinen luokittelumenetelmä, joka kuuluu ohjatun oppimisen alaluokkaan. Periaatteessa ja yksinkertaistettuna tukivektorikone erottelee kaksi luokkaa toisistaan lineaarisen tason avulla, siten, että kahden datajoukon väliin jää mahdollisimman suuri marginaali eikä yksikään datapiste jää marginaalitasojen sisäpuolelle. Tukivektorikonetta voidaan laajentaa useammankin luokan tai datajoukon erottamiseen, mutta tämä vaatii sen, että jokaisella datajoukolle koulutetaan oma luokittelija, joka erottaa datapisteet tästä luokasta omaksi joukokseen ja erilleen muista. [3]

Tukivektorikonetta voidaan käyttää sekä regressio-ongelmien kuin myös luokitteluongelmien ratkaisemiseen. Siitä huolimatta, tukivektorikonetta on käytetty eniten vaikeiden luokitteluongelmien ratkaisemiseen. [11]

Alla olevassa kuvassa 2 havainnoidaan tilannetta, jossa tukivektorikoneella on luokiteltu datajoukko siten, että datajoukkojen välille muodostuu mahdollisimman suuri marginaali.



Kuva 2. Tukivektorkonetta havainnollistava kuva.

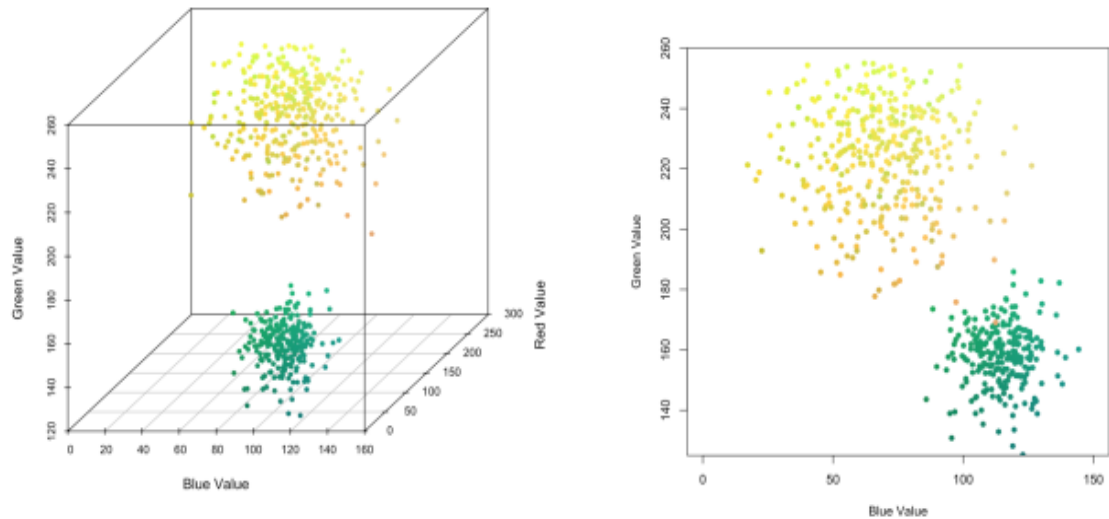
2.6 Itseohjautuvakartta

Itseohjautuvakartta eli Self-Organizing Map (SOM) on suomalaisen Teuvo Kohosen alun perin 1980-luvulla kehittämä ohjaamattomaan oppimiseen perustava neuroverkkomalli. Itseohjautuvankartan avulla pystytään yksinkertaisesti visualisoimaan monimutkaisia data-aineistoja, joissa on epälineaarisia riippuvuussuhteita havaintojen välillä. Itseohjautuvankartan avulla nämä pystytään esittämään yleensä kaksiulotteisesti, jossa samanlaiset datajoukon mallit ovat lähellä toisiaan ja vastaavasti erilaiset datajoukon pisteet ovat kaukana toisistaan kartalla. Tämän mallin idea onkin yksinkertaistaa vaikeasti analysoitavia datajoukkoja kuitenkin muuttamatta havaintojen välisiä suhteita. [12]

Itseohjautuvakartta on ollut erittäin kätevä sovellus prosessien, laitteiden ja systeemin monitoroimisessa eri teollisuuden aloilla. Tekniikan ja tieteen aloilla yleisesti on lukemattomia tehtäviä, joissa tutkimuskohteet pitää luokitella luontaisten ominaisuuksien perusteella, johon itseohjautuvakartta soveltuu hyvin. [13]

Kuvassa 3 vasemmalla on yksinkertaistettu kolmiulotteinen-mallinnus, jossa muuttujat ovat vihreä, sininen ja punainen väri. Vieressä oikealla on samoista havainnoista yksinkertaistettu kaksiulotteinen mallinnus, jossa muuttujat ovat vihreä ja sininen väri.

Havainnot edustavat joko keltaista tai vihreää väriä ja muodostavat kaksi klusteria. Opetuksen aikana itseorganisoiduvan kartan neuronit järjestäytyvät kahteen ryhmään, jotka vastaavat keltaisia ja vihreitä pisteitä. Kuten yllä mainittu, havainnot pystytään esittämään yksinkertaisemmin kaksiulotteisesti verrattuna kolmiulotteiseen esitykseen, kuitenkin sillä tavalla, ettei havaintojen väliset suhteet muutu. [14]



Kuva 3. Itseohjautuvakarttaa havainnollistava kuva.

2.7 Klusterointi

Klusteroinnilla (englanniksi clustering) on tarkoitus opettaa algoritmi tunnistamaan datajoukosta samankaltaiset kohteet (klusterit) ja luokittelemaan nämä kohteet omiin ryhmiinsä. [15] Teoriassa saman ryhmän klustereilla pitäisi olla samankaltaisuuksia, kun taas eri ryhmien klustereilla pitäisi olla eroavaisuuksia toisiinsa nähden. Klusterointi on hyvä esimerkki ohjaamattomasta oppimisesta, missä algoritmi oppii tunnistamaan datajoukosta samankaltaiset yksilöt ja luokittelemaan ne samankaltaisen yksilöiden kanssa samaan ryhmään. [16] Kuitenkaan algoritmi ei osaa nimetä yleensä ryhmiä sen tarkemmin, jolloin tarvitaan ihmistä nimeämään ryhmät valmiiksi ja tämän jälkeen klusterointialgoritmi pystyy luokittelemaan datajoukon pisteet järkevästi. [4]

Klusterointia käytetään yleisesti tilastollisen datan analysointiin ja se on käytössä monilla eri tieteen aloilla. [16] Esimerkkinä tästä käy ruokakaupan ostosdatan analysointi. Tässä tapauksessa voidaan luokitella asiakkaat tiettyihin asiakaskuntiin vaikkapa

ostoskäyttäytymisen, iän tai kaupassa asioimisajankohdan perusteella. Koneoppimisalgoritmin avulla pystyy sitten kohdentamaan asiakasryhmässä suosittuja tuotteita saman asiakasryhmän asiakkaille. [4]

Yksi tunnetuimmista ja yleisemmin käytössä olevista klusterointialgoritmeista on ”K-Means+ Clustering”. Tässä menetelmässä ideana on valita ensiksi, kuinka moneen ryhmään tai luokkaan algoritmi tulokset luokittelee. Seuraavaksi algoritmi laskee etäisyyden jokaisen datajoukon pisteen ja ryhmän keskustan väliltä ja luokittelee pisteen ryhmään, johon etäisyys on pienin. Näiden luokiteltujen pisteiden perusteella lasketaan ryhmän keskus uudestaan niin kauan, ettei ryhmän keskus juurikaan siirry iteraatioiden välissä. [16]

Menetelmän hyvä puoli on menetelmän nopeus, joka johtuu vähäisestä laskennan määrästä. Huonoja puolia vastaavasti tällä menetelmällä on muun muassa se, että ensiksi täytyy päättää, kuinka monta ryhmää ja/tai luokkaa datajoukosta löytyy, eikä tämä ole aina niin yksinkertaista. Väärin valitut ryhmät voivat aiheuttaa tuloksissa virheitä tai tulokset eivät ole sitä mitä haettiin. [16]

3 OPPIMISEN OPTIMOINTI

3.1 Kohdefunktio

Johdannossa käytiin lyhyesti läpi kohdefunktion merkitystä koneoppimisessa. Kohdefunktiolla siis tarkoitetaan koneoppimisalgoritmin kohdetta eli mihin se pyrkii toiminnallaan. Kohdefunktiolla on merkittävä vaikutus siihen, kuinka koneoppimisesta saadaan mahdollisimman hyvin optimoitua ja kuinka algoritmi toimii parhaiten. Kohdefunktio on optimoinnin kannalta avainasemassa, sillä väärällä kohdefunktiolla koneoppimisalgoritmi ei välttämättä toimi kuten halutaan tai se toimii muuten vain heikosti verrattuna paremmin valittuun kohdefunktioon. [1]

Kohdefunktion valintaan liittyy läheisesti myös se, minkälaista opetusdataa on käytössä ja riippuen kohdefunktion monimutkaisuudesta, vaaditun opetusdatan määrä lisääntyy lineaarisesti kohdefunktion monimutkaisuuden kanssa. Eli mitä monimutkaisempi kohdefunktio luodaan algoritmille, sitä enemmän vaaditaan opetusdataa ja aikaa opetusvaiheeseen. [1]

Myöskin johdannossa käytetty termi ”yleistävyys” liittyy läheisesti kohdefunktioon ja sen valintaan. Oikeanlaisen kohdefunktion valinta algoritmille parantaa algoritmin kykyä yleistää tilanteita ja sitä, kuinka se toimii tilanteissa, jotka eivät tulleet vastaan harjoittelun aikana. [1]

3.2 Optimointimenetelmät

Optimoinnin kannalta dataa ei ikinä voi olla liikaa koneoppimisalgoritmia opetettaessa, joten optimoinnin kannalta on tärkeää hankkia tarpeeksi data-aineistoa, jotta algoritmi voidaan optimoida mahdollisimman hyvin. [3] Tämän kappaleen alaotsikoissa käydään läpi optimointimenetelmiä, jotka ovat tärkeitä koneoppimisen parissa.

3.2.1 Particle Swarm Optimization

Particle Swarm Optimization (jäljempänä PSO) on iteratiivinen optimointimenetelmä, joka perustuu osittain kuitenkin sattumaan. Se on erityisen tehokas, kun pitää optimoida koneoppimisalgoritmi, jolla pyritään ratkaisemaan epälineaarisia ongelmia. [17]

PSO:lla on yhteyksiä luontoon, sillä esimerkiksi mehiläisillä on PSO:n tyylinen toimintatapa. Jos yksi mehiläisistä löytää paikan mistä löytyy paljon siitepölyä ja nektaria, mehiläinen informoi muita mehiläisiä pesässä, jotka pian löytävät samalle paikalle, tosin eri reittejä pitkin. Tässä samalla saattaa löytyä parempia paikkoja, joista taas informoidaan muita mehiläisiä. Lopputulema tässä on se, että koko ajan löytyy parempia paikkoja mehiläisille ikään kuin vahingossa johtuen erilaisista reiteistä kohti aikaisempaa päämäärää. [17]

Tämä sama pätee PSO:n tekniikkaan, jokainen partikkeli pystyy kommunikoimaan toisille ja kertomaan oman informaationsa parhaasta ratkaisusta minkä se tietää. Näin muut partikkelit pystyvät hyödyntämään eri lähteistä tulevaa informaatiota ja tekemään omat ratkaisunsa, jotka ne taas voivat jakaa eteenpäin. Näin koneoppimisalgoritmi optimoituu pikkuhiljaa partikkeleiden yhteistyön avulla. [17]

Parhaassa tapauksessa nämä partikkelit eivät viestittele kaikki täysin keskenään, koska silloin ne päätyvät samaan lopputulokseen ja optimointi loppuu siihen. Toisaalta jos viestintä partikkeleiden välillä on liian vähäistä, tieto ei välity kunnolla hyvästä informaatiosta ja optimointi ei tällöin ole parasta mahdollista. [17]

Partikkeleiden määrä eroaa myös mehiläisparvesta huomattavasti. Mehiläisparvesta puhuttaessa puhutaan normaalisti noin 20000 yksilöstä, PSO:n tapauksessa 20-40 partikkelin määrä on riittävä. PSO:n yksinkertaistetussa mallissa menetelmä eroaa myös mehiläisparven toiminnasta huomattavasti. Siinä missä mehiläiset yrittävät etsiä monia hyviä ratkaisuja, partikkelit pyrkivät löytämään ainoastaan parhaan mahdollisen ratkaisun. [17]

3.2.2 Geneettinen algoritmi

Geneettinen algoritmi (GA) on optimointimenetelmä, joka perustuu evoluution mekanismeihin ja se on heuristinen optimointimenetelmä eli etsivä optimointimenetelmä. Geneettinen algoritmi sopii optimointimenetelmäksi tehtäviin, joissa pitää etsiä erilaisia ratkaisuja. Kuten evoluutiokin, geneettinen algoritmi perustuu osittain sattumaan. [18]

Jokaista geneettistä algoritmia yhdistää samanlaiset pääpiirteet, joita ovat populaatio, joka on joukko alkioita ratkaisemaan haluttu ongelma. Tämän populaation sisällä tapahtuu valinta kuten evoluutiossakin; parhaimmat yksilöt selviävät ja lisääntyvät risteyttämällä valitut yksilöt. Tässä tapauksessa jonkin valintakriteerin mukaan valitaan parhaat alkiot joukosta ja risteytetään nämä keskenään. Jotta optimointi olisi mahdollisimman tehokasta, tehdään mutaatioita joissakin alkioiden ominaisuuksissa. Näiden yksilöiden jälkeläisistä syntyy taas uusi sukupolvi. . Jokainen sukupolvi korvaa aina edellisen sukupolven ja tätä sykliä jatketaan, kunnes saavutetaan haluttu ratkaisu selvitetävään ongelmaan. Tämä menetelmä perustuu myös samalla tavalla iterointiin mitä ylempänä mainittu PSO. Tässä tapauksessa jokainen uuden sukupolven syntyminen on yksi iteraatiokerta. [18]

Geneettisen algoritmin onnistuminen riippuu hyvin paljon alkupopulaation koosta ja risteytyksen ja mutaation valintakriteereistä ja näiden todennäköisyyksistä. Vaikkakin geneettinen algoritmi perustuu osittain sattumaan, tämä on erittäin tehokas optimointimenetelmä, kun kyseessä on vaikeasti ratkaistavat optimointitehtävät. [18]

3.2.3 Differential Evolution

Differential Evolution algoritmit (DE) ovat myös heuristisia optimointimenetelmiä eli etsiviä optimointimenetelmiä, kuten kappaleessa 3.2.2 läpi käyty geneettinen algoritmi. Niissä on samankaltaisuuksia muutenkin geneettisten algoritmien kanssa, kuten tärkeimmät operaattorit, joita ovat populaation sisällä tapahtuva valinta, risteyttäminen ja mutaatio. Näiden lisäksi molempien toiminta perustuu iteratiiviseen prosessiin. Pääperiaatteeltaan niiden toiminta on samanlaista kuin geneettisissä algoritmeissa, mutta on niissä myös eroavaisuuksia. Differential Evolution algoritmeissa on todettu toimivaksi käyttää ”yritys ja erehdys”-metodia, jonka avulla löydetään sopivat parametrit esimerkiksi risteytysten määrälle. Eli kokeillaan esimerkiksi jotain parametrien arvoa ja

sillä saadaan tietää toimiiko se siinä kohtaa evoluutiota vai ei. Haittapuolena tässä on aikaa vievät ja kalliit laskennalliset kustannukset. Huolimatta tästä DE on erittäin tehokas optimointimenetelmä. [19]

Kuitenkin, jotta DE:llä saavutetaan parhaat tulokset, täytyy käyttää erilaisia strategioita eri vaiheissa evoluutiota. Tästä esimerkkinä on parametrien vaihtaminen eri vaiheissa prosessia. Joissakin vaiheissa evoluutiota parametrien muuttaminen, kuten aiemmin mainittu risteytysten määrä, voi tehostaa optimointia. Esimerkiksi nostamalla risteytymisten määrää tietyssä vaiheessa prosessia optimointiprosessi nopeutuu. Tämäkin omalta osalta lisää laskennallisia kustannuksia, joka pitää ottaa huomioon. [19]

DE:n optimointitehokkuus riippuu myös samoista asioista kuin geneettisen algoritmin tehokkuus. Siinä tärkeitä tekijöitä ovat esimerkiksi populaation koko, joka valitaan alussa. Liian laaja populaation koko vaikeuttaa optimointia. Toisaalta myös liian suuri mutaatioiden määrä risteyttämisen yhteydessä voi johtaa ei-haluttuihin tuloksiin optimoinnissa, aivan kuten geneettisten algoritmien kohdalla. Differential Evolution optimointimenetelmät ovat käytössä algoritmien optimoinnissa muun muassa konetekniikan ja hahmontunnistuksen osa-alueilla. [19]

3.3 Optimoinnin ongelmat

Käydään läpi tämän kappaleen alaotsikoissa koneoppimisen erilaisia ongelmia ja miten ne vaikuttavat koneoppimisalgoritmien optimointiin.

3.3.1 Ylioppiminen

Ylioppimisella (englanniksi variance) tarkoitetaan sitä, kun malli kuvaa harjoitteluvaiheessa käytettyä opetusdataa ja siinä esiintyvää vaihtelua liian tarkasti, eikä yleisty tarpeeksi hyvin, jotta mallin ennustavuus olisi tarpeeksi hyvä uudelle datajoukolle. Tätä tapahtuu juuri silloin, kun näytejoukko on entuudestaan tuntematon algoritmille. [3] Esimerkkinä tästä, kappaleessa 2.2 käytiin läpi päätöksentekopuita ja niissä esiintyvä mallin monimutkaisuus johtaa yleistyskyvyn heikkenemiseen eikä haluttuun lopputulokseen välttämättä päästä. Tämä johtuu juuri ylioppimisesta. [7]

Syitä tähän ongelmaan voivat olla liian kapea koulutusdata tai mallin liiallinen monimutkaisuus. Ylioppiminen voidaan ratkaista lisäämällä koulutusdataa tai yksinkertaistamalla mallia. Yleensä algoritmi on joko ali- tai ylioppinut, eikä niin sanotusti täydelliseen lopputulokseen pyrkiminen ole edes kannattavaa algoritmin tehokkuuden näkökulmasta. [3]

3.3.2 Alioppiminen

Alioppimisella (englanniksi bias) vastaavasti tarkoitetaan sitä, ettei algoritmi ole pystynyt sisäistämään opetusdatassa esiintyvää vaihtelua, eikä täten pysty kuvaamaan sitä riittävän hyvin uusissa näytteissä, josta johtuu, ettei se myöskään pysty ennustamaan uusien näytteiden käyttäytymistä riittävän tarkasti. Malli on hyvin yksinkertainen tässä tapauksessa, koska vaihtelua ei juuri esiinny. [3]

Syitä, jotka johtavat alioppineeseen mallin voivat olla esimerkiksi parametrien vähäinen määrä mallissa tai mallin epäsopivuus tiettyyn ongelmaan. Alioppimiseen ei riitä koulutusdatan lisääminen, sillä alioppiminen johtuu mallin huonosta oppimiskyvystä, joten alioppimisesta päästään kehittämällä ja optimoimalla malli paremmin. [3]

4 SOVELLUKSET

Tässä kappaleessa käydään läpi koneoppimisen ja tekoälyn avulla saavutettuja sovelluksia, joista osa on hyvinkin tuttuja arkipäiväisessä elämässämme vaikkei sitä aina tule ajatelleeksi.

4.1 Hahmontunnistus

Hahmontunnistuksella tarkoitetaan koneoppimiseen liittyviä algoritmeja ja menetelmiä, joiden avulla mittausdatasta, kuten kuvista pystytään tunnistamaan haluttuja säännönmukaisuuksia, kuten kasvoja tai liikennemerkkejä. Toinen esimerkki hahmontunnistuksesta on vaikkapa postinumeroiden tunnistaminen automaattisesti kirjekuoresta. Hahmontunnistus perustuu siihen, että ympäristöstä saadaan mittausdataa erilaisilla sensoreilla mittaamalla, kuten kameralla saadaan kuvaa tai mikrofoniin ääniä. [20]

Otetaan tarkempaan tarkasteluun kuvantunnistus ja vielä tarkemmin kasvojen tunnistus kuvista. Kuvantunnistuksella tarkoitetaan juurikin tekoälyn ja koneoppimisalgoritmien käyttämistä kuviin ja tiettyjen säännönmukaisuuksien tunnistamiseen näistä kuvista. Perusidea kasvojen tunnistuksessa kuvista on löytää ihmisille tyypilliset kasvonpiirteet kuten silmät, nenä ja suu, joiden perusteella algoritmi päättää onko kyseessä kasvot vaiko ei. Koneoppimisalgoritmeista voidaan käyttää kasvontunnistukseen muun muassa neuroverkkoja, tukivektorikoneita ja Naiivia Bayesin luokitinta. [20]

Haasteita kasvojen tunnistamisessa on tietenkin kuvassa esiintyvät muut asiat kuten tausta ja esineet kuvassa. Mutta myös kasvojen ilme, ihmisen asento ja erityisesti kasvojen asento sekä kuvassa esiintyvät olosuhteet (esimerkiksi auringonvalon määrä) voivat olla haasteita koneoppimisalgoritmille. Muutenkin ihmisten kasvot ovat erilaisia koon, muodon ja värinkin puolesta, joten opetusdatan määrän täytyy olla laaja kattaus erilaisia kuvia. [21]

Kasvojen tunnistus kuvasta tekoälyn avulla lienee monille se tunnetuin muoto tekoälyn käytöstä nykymaailmassa. Monissa elokuvissa ja sarjoissa esimerkiksi poliiseilla on käytössä kasvontunnistus.

4.2 Lääketieteellinen diagnoosi

Nykyään on tärkeää pystyä luokittelemaan ihmiset korkean ja pienen riskiluokan välille esimerkiksi jonkin tietyn syöpädiagnoosin suhteen. Koneoppimisen avulla tämä on helpottunut, kun sillä pystytään karsimaan suurista datamääristä suurella todennäköisyydellä ihmiset riskiluokkiin, joihin he kuuluvat tai ennustamaan kuinka todennäköistä on selvitä jostain syövästä, jos tietyt oireet esiintyvät tai potilaalla on jotain alttiuteen johtavia ominaisuuksia. Esimerkiksi naisten rintasyövän parantumisenusteessa tärkeitä tekijöitä ovat kasvaimen koko, etäpesäkkeiden määrä ja naisen ikä. [22]

Koneoppimisalgoritmeilla pystytään tunnistamaan tiettyjä kaavoja ja suhteita monimutkaisista data-aineistoista ja pystytään tehokkaasti ennustamaan kuinka esimerkiksi sädehoito tehoaa tietynlaiseen syöpään ja millä todennäköisyydellä ihminen selviää syövästä ja millä todennäköisyydellä syöpä uusiutuu tulevaisuudessa. Myöskin koneoppimisalgoritmien avulla on pystytty selvittämään tiettyyn syöpään johtavia riskitekijöitä ja täten ennustamaan alttius algoritmin avulla. Koneoppimisalgoritmien avulla data-aineistoista, joita on kerätty potilaista vuosien varrella, on pystytty löytämään yhtäläisyyksiä, joita ei välttämättä muuten olisi löytynyt. [22]

Joitain koneoppimisalgoritmeja on käytetty melkein kolmen vuosikymmenen ajan syöpien seulontaan. [22]

4.3 Sähköpostin roskapostisuodatin

Roskapostisuodattimilla on todellinen tarve, sillä jopa 77% kaikesta maailman sähköpostiliikenteestä on roskapostia. Koneoppimisella on saavutettu automaattisesti roskapostin tunnistavia ja suodattavia roskapostisuodattimia, ja ne ovat käytössä suurilla palveluntarjoajilla kuten Google, Yahoo ja Outlook. Nämä palveluntarjoajat käyttävät koneoppimisalgoritmeihin perustuvia roskapostisuodattimia. Esimerkiksi neuroverkkoihin perustuvat roskapostinsuodattimet pystyvät itsenäisesti oppimaan koko ajan paremmin erottamaan roskapostia käymällä läpi aikaisemmin saapunutta roskapostia ja opetusdataa sekä pystyvät ajan mittaan adaptoitumaan tunnistamaan ja suodattamaan uudenlaisia roskapostiviestejä. [23]

Käytännössä koneoppimisalgoritmit etsivät samankaltaisuuksia viestien väliltä viestien rakenteesta, sanamäärästä, sanoista ja lähettäjistä sekä vastaanottajista ja tämän perusteella lajittelee viestin joko roskapostiksi tai tärkeäksi viestiksi (spam or ham). Googlen Gmailin roskapostisuodatin pystyy myös luomaan uusia sääntöjä koskien roskapostia tullen näin entistä tehokkaammiksi ja se pystyykin jo 99.9% tarkkuudella suodattamaan roskapostin tärkeistä viesteistä. [23]

4.4 Mineraalien erottaminen hiekan seasta konenäön avulla

Konenäkö, joka toimii koneoppimisalgoritmeilla on pystytty valjastamaan mineraalijyvien löytämiseen ja niiden laskemiseen hiekan seasta. Tällä menetelmällä on saavutettu jo sellainen taso, että konenäkö erottaa ja pystyy luokittelemaan jopa 90% tarkkuudella mineraalijyvät hiekanjyvien seasta. Normaalisti prosessi on hoidettu optisella mikroskoopilla tai elektronimikroskoopin avulla, mutta koska prosessi on aikaa vievä, turhauttava ja kallis, on koneoppimisen avulla saavutettu huomasti edistystä tämän hoitamiseen. Nyt prosessi hoidetaan näyttämällä konenäkölle optisella mikroskoopilla otettuja kuvia, ja tämä opetusdatan perusteella tunnistaa mineraalijyvät, eikä ihmisen tarvitse käsin alkaa etsimään niitä. [24]

Pelkästään koneoppimisalgoritmeilla ei saavutettu niin hyviä tuloksia kuin yhdistämällä konenäkö toimimaan koneoppimisalgoritmeilla. Konenäkö tässä tapauksessa valjastettiin käyttämään yksinkertaisia luokittelualgoritmeja kuten päätöspuumenetelmää, jota käytiin läpi kappaleessa 2.2. [24]

Tämä menetelmä on kuitenkin todella uusi, mutta silti sitä pidetään erittäin lupaavana ja innovaatisena menetelmänä ja voidaan mahdollisesti soveltaa tulevaisuudessa muihinkin vastaaviin tehtäviin. [24]

5 TULEVAISUUDEN VISIOITA TEKOÄLYSTÄ

Tekoäly on ollut kanssamme jo pitkään erinäisissä sovelluksissa, mutta vasta viimevuosina tekoäly ja siihen läheisesti liittyvä koneoppiminen on ottanut erittäin suuria harppauksia eteenpäin. Hyvänä esimerkkinä Boston Dynamics nimisen yhtiön kehittämät robotit, jotka kykenevät hyvinkin vaativiin tehtäviin ja ovat uskomattoman kehittyneitä robotteja. Ne pystyvät esimerkiksi väistämään esteitä ja kiipeämään niiden yli ihmismäisesti. Luultavammin he käyttävät Reinforcement Learningia robottien kouluttamisessa.

Tulevaisuudessa tekoäly on luultavasti vieläkin suurempi osa jokapäiväistä elämää, vaikka se on sitä jo nytkin. Aiemmin ei välttämättä edes ymmärtänyt kuinka paljon koneoppimisella on saatu aikaan mullistavia sovelluksia, kuten älykkäät karttasovellukset, jotka pystyvät ehdottamaan vaihtoehtoisia reittejä liikenteen sujuvoittamiseksi tai jokaiselle tuttu Facebook, joka yksilöi uutisvirran jokaiselle käyttäjälle tekoälyn avulla.

Tulevaisuudessa joukkoliikenne toimii mahdollisesti täysin automatisoidusti koneoppimisen avulla ja esimerkiksi lentokoneet eivät tarvitse pilottia. Aikajännettä tähän on vaikea arvioida, mutta sen voi olettaa tapahtuvan aivan lähitulevaisuudessa, sillä onhan Pohjois-Amerikassa testattu jo hyvän tovin itsenäisesti ajavia autoja.

Mutta tekoälyn yleistyminen luo myös tietynlaisia uhkakuvia, kuten sen että kuinka rajusti työttömyys lisääntyy ympäri maailmaa, jos suurin osa töistä on korvattavissa roboteilla tai muuten pystytään automatisoimaan. Jos näin tapahtuu, tulevaisuuden yhteiskunnat voivat näyttää täysin erilaisilta mitä nykypäivän yhteiskunnat ja voi olla, ettei ihmistä nähdä samanlaisena työtä tekevänä yhteiskunnan jäsenenä kuin mitä nykypäivänä jokaiselta työkykyiseltä odotetaan.

Toisaalta myös monet tiedemiehet ja tekoälyn kanssa kamppailevat asiantuntijat ovat varoitelleet tekoälyn monista mahdollisuuksista sodankäynnin kannalta ja varoitelleet tekoälyn valjastamisesta sotateollisuuden käyttöön. Kaikista perinteisin, mutta ei välttämättä niin todennäköinen uhkakuva on varmaankin tekoälyn kehittyminen ihmisen yläpuolelle niin, että se syrjäyttäisi koko ihmisrodun. Monet asiantuntijat ovat varoitelleet

tästä, mutta ajatus robottien tai tekoälyn vallankumouksesta, joka syrjäyttäisi ihmiset täysin ja alistaisi orjuuteen, kuulostaa liian absurdilta. Ennemmin tekoälyn ja robottien voi arvella tulevan ihmisten rinnalle ja koneoppimisella pystytään automatisoimaan teollisuudessa ja muillakin aloilla tehtäviä, jotka ovat työläitä ihmisille tai muuten vain tehostaa tuotantoa huomattavasti.

6 YHTEENVETO

Työssä oli tarkoituksena avata koneoppimista ja koneoppimismenetelmiä sekä siihen liittyviä sovelluksia. Koneoppiminen on kuitenkin terminä suhteellisen laaja käsite ja tässä työssä sitä pyrittiinkin avaamaan mahdollisimman laajasti ja selkokielisesti menemättä kuitenkaan liian syvälle tieteelliseen puoleen.

Koneoppimiseen liittyy paljon tärkeitä käsitteitä kuten esimerkiksi kohdefunktio, yleistävyys ja optimointi. Käsitteiden ymmärtäminen helpottaa ymmärtämään kuinka algoritmit toimivat ja miten ne oppivat. Koneoppimismenetelmät voidaan karkeasti jakaa kahteen alaluokkaan, joita ovat ohjattu ja ohjaamaton oppiminen, sekä näiden rinnalla on Reinforcement Learning, joka on periaatteessa välimuoto molemmista. Näitä menetelmiä on kuitenkin lukuisia ja tässä työssä käytiin läpi semmoiset koneoppimismenetelmät, jotka ovat olennaisia ja pyrittiin valitsemaan menetelmiä molemmista alaluokista. Menetelmistä käytiin läpi myös käytännön esimerkkejä selkeyttämään kuinka ne todellisuudessa toimivat. Koneoppimismenetelmien jälkeen paneuduttiin syvällisemmin, miten koneoppimisalgoritmien toimintaa optimoidaan muun muassa kohdefunktion valinnalla ja minkälaisia optimointimenetelmiä on olemassa ja miten nämä optimointimenetelmät toimivat. Vastapainoksi käytiin myös läpi mitä ongelmia liittyy oppimisen optimointiin.

Työssä käytiin läpi tarkemmin muutamaa koneoppimisella saavutettua sovellusta, joista pari oli tunnetumpaa ja toiset pari vähemmän tunnettuja. Sovellukset antavat konkreettisen kuvan mitä koneoppimisella on jo saavutettu sekä pientä esimakua mitä niillä voidaan mahdollisesti tulevaisuudessa saavuttaa. Viimeisenä on myös omaa pohdintaa tekoälyn tulevaisuudesta ja mahdollisista uhkakuvista, jotka ovat mahdollisia tekoälyn kehittyessä.

Jos aihetta haluaisi syvällisemmin käydä läpi, silloin pitäisi mennä matemaattisten mallien ja koodaamisen puolelle, joka menee tässä tapauksessa aihepiirin ulkopuolelle. Työssä kuitenkin käytiin vain pintaraapaisu koneoppimisesta, mutta kuitenkin sen verran kattavasti, että tämän työn luettuaan ymmärtää mistä koneoppimisesta ja siihen liittyvissä menetelmissä on kyse.

LÄHDELUETTELO

- [1] Mitchell, T., 1997. Machine Learning. New York (NY): McGraw-Hill, 414 s. ISBN 0-07-042807-7. s.2-3 & 7-8.
- [2] Roos, T., 2019. Johdatus tekoälyyn. [verkkodokumentti]. Helsinki: Helsingin yliopisto. Saatavissa: https://www.cs.helsinki.fi/webfm_send/932 (2) [viitattu 19.2.2019].
- [3] Eskelinen, M., 2018. Luku 14 – Koneoppiminen ja luokittelu [verkkodokumentti]. Jyväskylä: Jyväskylän yliopisto. Saatavissa: http://users.jyu.fi/~amjayee/TIES411/chapter_14.fi.pdf [viitattu 25.6.2019].
- [4] University of Helsinki, 2019. I. Koneoppimisen lajit [verkkodokumentti] Helsingin yliopisto. Saatavilla: <https://course.elementsofai.com/fi/4/1> [viitattu 18.01.2019].
- [5] ReaktorAI, 2018. Kannattaako kauppa. [verkkodokumentti]. Saatavissa: <http://kannattaakokauppa.fi/#/fi/> [viitattu 14.3.2019].
- [6] Haykin, S., 2009. Neural Networks and Learning Machines. Third Edition. McMaster University, Canada: Pearson, 936 s. ISBN: 978-0131471399 s.36-38.
- [7] Maimon, O., Rokach, L., 2008. Data Mining with Decision Trees: Theory and Applications. World Scientific Publishing. 244 s. ISBN 978-981-277-171-1 s. 1-19.
- [8] University of Helsinki, 2019. III. Naiivi Bayesin luokitin [verkkodokumentti]. Helsinki: Helsingin yliopisto. Saatavilla: <https://course.elementsofai.com/fi/3/3> [viitattu 13.04].
- [9] McCallum, A., Nigam, K., 1998. A Comparison of Event Models for Naive Bayes Text Classification [verkkodokumentti] Pittsburgh: Carnegie Mellon University. Saatavissa: <http://www.cs.cmu.edu/~knigam/papers/multinomial-aaaiws98.pdf> [viitattu 24.7.2019] 8 s.

- [10] DeepAI. Neural Network [verkkodokumentti]. Saatavissa: <https://deepai.org/machine-learning-glossary-and-terms/neural-network> [viitattu 14.6.2019].
- [11] Haykin, S., 2009. Neural Networks and Learning Machines. Third Edition. McMaster University, Canada: Pearson, 936 s. ISBN: 978-0131471399 s.268-270.
- [12] Kohonen, T., 2005. The Self-Organizing Map (SOM). [verkkodokumentti] Helsinki: Helsinki University of Technology. Saatavissa: <http://www.cis.hut.fi/projects/somtoolbox/theory/somalgorithm.shtml> [viitattu 11.4.2019].
- [13] Kohonen, T., 2014. MATLAB Implementations and Applications of the Self-Organizing Map. Helsinki, Finland: Unigrafia Oy. ISBN: 978-952-60-3678-6.
- [14] Ng, A., Soo, K., 2017. Self-organizing Maps Tutorial [verkkodokumentti]. Saatavissa: <https://annalysin.wordpress.com/2017/11/02/self-organizing-map/> [viitattu 1.3.2019].
- [15] Le, J., 2016. The 10 Algorithms Machine Learning Engineers Need to Know. [verkkodokumentti] Saatavissa: <https://www.kdnuggets.com/2016/08/10-algorithms-machine-learning-engineers.html/2> [viitattu 28.3.2019].
- [16] Seif, G., 2018. The 5 Clustering Algorithms Data Scientists Need to Know [verkkodokumentti]. Saatavissa: <https://towardsdatascience.com/the-5-clustering-algorithms-data-scientists-need-to-know-a36d136ef68> [viitattu 28.3.2019].
- [17] Clerc, M., 2006. Particle Swarm Optimization. London: ISTE Ltd, 243 s. ISBN 978-1-905209-04-0 Luvut 1 & 2
- [18] Mitchell, M., 1998. An Introduction to Genetic Algorithm. Fifth Printing. Cambridge, Massachusetts: The MIT Press, 158s. ISBN 0-262-63185-4 s.7-11 & s. 135

- [19] Qin, A.K., Huang, V.L., Suganthan, P.N., 2009. Differential Evolution Algorithm With Strategy Adaptation for Global Numerical Optimization. *IEEE Transactions on Evolutionary Computation*, 13 (2), S. 398-417
- [20] Seppänen, T., 2019. 1. Johdatus tilastolliseen hahmontunnistukseen [verkkodokumentti]. Oulu: Oulun yliopisto Saatavissa: <http://www.ee.oulu.fi/research/tklab/courses/521497S/pruju/Chapter1.pdf> [viitattu 14.08.2019]
- [21] Yang, M-H., Kriegman, D.J., Ahuja, N., 2002. Detecting Faces in Images: A Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. Volume 24 (1), S. 34-58 Saatavissa: <http://dx.doi.org/10.1109/34.982883> [Viitattu 10.8.2019]
- [22] Kourou, K., Exarchos, T.P., Exarchos, K.P., Karamouzis, M.V., Fotiadis, D.I., 2019. Machine learning applications in cancer prognosis and prediction. *Computational and Structural Biotechnology Journal*, Volume 13, S. 8-17. Saatavissa: <https://doi.org/10.1016/j.csbj.2014.11.005>. [Viitattu 14.9.2019]
- [23] Dada, E.G., Bassi, J.S., Chiroma, H., Abdulhamid, S.M., Adetunmbi, A.O., Ajibuwa, O.E., 2019. Machine learning for email spam filtering: review, approaches and open research problems. *Heliyon*, Volume 5 (6). Saatavissa: <https://doi.org/10.1016/j.heliyon.2019.e01802>. [Viitattu 14.9.2019]
- [24] Maitre, J., Bouchard, K., Bédard, L.P., 2019. Mineral grains recognition using computer vision and machine learning. *Computers & Geosciences*, Volume 130, 2019, S. 84-93.